IN THE TITLE:

Please revise the title of the application to read as follows:

--MANAGING A CLUSTER OF NETWORKED RESOURCES AND RESOURCE GROUPS USING RULE-BASED CONSTRAINTS IN A SCALABLE CLUSTERING ENVIRONMENT--

IN THE SPECIFICATION:

Please delete the two (2) paragraphs beginning on Page 4, Line 13 through Page 5, Line 11 and replace them with the revised three (3) paragraphs, as follows:

~~To achieve the above objectives, and to overcome the above-discussed difficulties of the prior art, the present invention take an approach that is different from the traditional resource management approach. In this approach, resources are considered as services whose availability and quality-of-service depends on the availability and the quality-of-service provided by one or more other services in the cluster. For this reason, the cluster and its resources may be represented by two dimensions. The first dimension captures the semi-static nature of each resource; e.g., the type and quality of the supporting services needed to enable its services. Typically, these requirements are defined (explicitly or implicitly) by the designers of the resource or the application. These may be further qualified by the cluster administrators. These are formalized as simple rules that can be dynamically and programatically evaluated, taking into account the current state of the cluster. The second dimension is the dynamic state of the various services provided by the cluster. The dynamic changes are captured by events. Finally, all the coordination and mapping is done at a logically centralized place, where the events are funneled in and the rules are evaluated. This helps in isolating and localizing all the heterogeneity and associated complexity. By separating the dynamic part (the events) from the semi-static parts (the rules), and combining these in a systematic manner only when needed, the desired level of automation in the coordination and mapping of resources and services can be achieved.~~

~~While the general principles outlined above are fairly straightforward, there is a nontrivial amount of complexity in managing the choreography. To show the proof of concept, we have designed and implemented a system called Mounties based on the above described general principles. The Mounties architecture itself is composed of multiple components, a primary component being the modeling and decision making engine. The remaining components together form an active and efficient resource management layer between the actual cluster resources and the decision-making~~

~~component. This layer continuously transports the state information to the decision maker and commands from the decision maker to the cluster resources, back and forth in a fault-tolerant manner.~~

--An embodiment of the invention provides a method of managing a cluster of networked resources and resource groups using rule-based constraints in a scalable clustering environment. This method comprises the step of building a globally optimal cluster configuration of said networked resources in accordance with said rule-based constraints and a current state of said resources, including identifying for each of the resources and resource groups an availability and quality of service, which are determined by dependencies among the resources and resource groups, resource equivalency, user preferences, constraints on the resources and network policies. The method comprises the further steps of bringing said cluster of networked resources on-line in a systematic manner, given the current state of each of the resources and resource groups, and their dependencies, user preferences, constraints on the resources, and network policies; and with said cluster of networked resources on-line, determining dynamic dependencies of and configuration information about said cluster of networked resources (i) statically at said step of building and said step of bringing said cluster of networked services online and (ii) dynamically during cluster operation in accordance with said rule-based constraints.

This method comprises the further steps of supporting startup, operation and shutdown of said cluster of networked resources according to current policies, and system events, and said rule-based constraints; separating said dependencies among resources and resource groups, user preferences, constraints among the resources, system events, and current policies into (i) a first, static rules based group and (ii) a second, dynamically changing events based group, wherein said first group captures the static resources, including, for each resource, a type and quality of the supporting resources needed to enable said each resource, wherein said step of separating is implemented according to said rule-based constraints; and combining said first and second groups in a systematic manner only when needed to build the said globally optimal cluster configuration, and only when needed during operation to modify and realign the current state of said cluster to said globally optimal cluster configuration, or an alternative globally optimal in view of said

current policies, said system events and said rule-based constraints. In this embodiment, the providing step includes providing a preprocessor module and a postprocessor module; the preprocessor module includes a preprocessor entry queue, the optimizer module includes an optimizer input queue, and the postprocessor module includes a postprocessor input queue.

In this embodiment, the method comprises the further step of creating a preprocessor task to implement a decision to reallocate a resource or a resource group, wherein said preprocessor task is provided to the entry queue of the preprocessor module; and wherein said preprocessor task is an object having an entry method that, when invoked, results in the preprocessor task being executed, and execution of the preprocessor task results in either a postprocessor task being provided in the postprocessor input queue, an optimizer task being provided in the optimizer input queue, or both. This method comprises the further steps of scheduling the postprocessor task by an invocation of the entry method associated with the postprocessor task; scheduling the optimizer task by an invocation of the entry method associated with the optimizer task; and executing the optimizer task results in a postprocessor task provided in the postprocessor input queue.--